Assignment #7 - Cloud Chat App

Introduction

The assignment consisted of changing the previous Chat App so that the app could send a list of messages and receive a list of messages and clients from a web server through http requests. The also manages the *offline state*, showing the messages that are pending and then automatically uploading these messages when connectivity is available.

Development

All the steps given in the assignment were followed and the class slides were used as guidelines for project organization (packaged and classes' names conventions). The rubric was fulfilled and also used as a guideline of the most important requirements for the system.

Every time the SEND button is pressed, the written message is saved to the local database. The app UI shows this new message with the status *pending*. When this message is sent to the server and returned with a sequential number, it is updated in the database and the UI shows it with the status *sent*. The messages sent by the user will display the name *me* above the message. The messages received from other users connected to the server will have their names above the content.

The system uses and recurrent event (every 10 sec) to synchronize any pending messages (that might be waiting due some previous connectivity problem) with the server and get new messages as well as an updated list of clients. These information obtained from the server will then be inserted/updated in the local database.

Testing the App

In order to test the app, it is necessary to first run the server first. The command used to run the server during the tests was: **java -jar ChatServerSimple.jar localhost 81**

The above command sets the server name and port. It is important to keep in mind that the port used in the command must be the same configured in the app.

With the app opened, it is necessary to go to the settings first (the cog icon in the menu) and set the server address and username. The address must be set according with the settings used to run the server. The address used during the tests was: **http://10.0.2.2:81**

After the address and username are set, returning the the main screen will make a request to the server, registering the new user. This action will executed only once, as the app registration key and user id will be stored in the shared preferences of the app. After a successful registration, the “SEND” button becomes enabled and it is possible to send messages to the server.

Conclusion

The project APK can be found at the directory ***ChatServerWebServer\app\build\outputs\apk***

The video of the ChatApp with tasks running in the same process is available at the root folder with the name of ***simple_cloud_chatapp.swf***.